Est-ce que ça s'arrête ?

Pierre Lescanne

Est-ce que ça s'arrête ?

Le 10 juillet 2009, par **Pierre Lescanne**
Professeur d'informatique à l'ENS de Lyon (**page web**)

Nous présentons divers problèmes de terminaison, dont la solution est plus ou moins facile et nous évoquons les applications de la terminaison en logique mathématique et en informatique.

Des petits cailloux

Des rouges, des bleus

IMAGINEZ un alignement de petits cailloux colorés rouges et bleus qu'une *petite main* s'amuse à modifier. Elle le fait suivant certaines règles précises ; elle considère une petite zone et y déplace des cailloux mais elle peut aussi en ajouter. Elle peut, par exemple, le faire en appliquant la règle suivante :

règle 22 donne 22

Ça veut dire que quand il y a deux cailloux rouges suivis de deux bleus, la *petite main* échange les rouges et les bleus. Avec cette règle, la *petite main* peut faire les transformations suivantes :

Exemple pour 22 donne 22

On voit que, quoi qu'elle fasse, elle ne peut pas déplacer les cailloux indéfiniment ; en effet, le nombre de cailloux ne change pas, et même plus précisément le nombre de cailloux bleus et le nombre de cailloux rouges ne changent pas. Mais ce qui change, c'est le fait que les cailloux bleus soient déplacés vers la gauche et il y aura un moment où la *petite main* ne pourra plus déplacer les cailloux bleus vers la gauche !

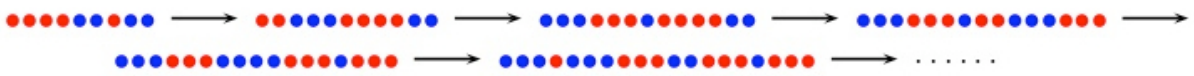
En revanche, si la *petite main* s'autorise à ajouter aussi des cailloux suivant la règle

alors elle ne s'arrêtera pas, car le motif constitué de deux boules rouges suivies de quatre boules bleues se reproduit en lui-même indéfiniment. Plus précisément, cela donne

example 22 donne 44

Si la *petite main* s'autorise à déplacer et à ajouter des cailloux, un bleu et un rouge, suivant la règle

cela donne par exemple :



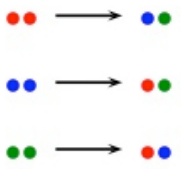
Va-t-elle s'arrêter de partant de la configuration ci-dessus ? Va-t-elle s'arrêter quelle que soit la configuration de départ ? [1]

La terminaison uniforme

Dans cet article, nous nous intéressons à la *terminaison uniforme*, on dit aussi l'*arrêt*, c'est-à-dire que nous tentons de répondre à la question « Le procédé de transformation s'arrête-t-il dans tous les cas ? » Un procédé s'arrête quand il n'y a plus rien à faire. Dans notre cas des alignements de cailloux, on s'arrête quand en cherchant partout dans l'alignement on ne trouve pas de configuration de cailloux qui puisse être transformée par la *petite main* en utilisant l'une des règles. La terminaison [2] est dite « uniforme » parce dans le cas où il y a plusieurs transformations possibles, on n'en privilégie aucune.

Des rouges, des bleus, des verts

Maintenant supposons qu'il y ait aussi des cailloux verts et les règles :



Autrement dit, deux rouges donnent un bleu suivi d'un vert, deux bleus donnent un rouge suivi d'un vert, deux verts donnent un rouge suivi d'un bleu.

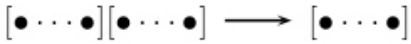
Est-ce que ça va s'arrêter [3]. Ou non ?

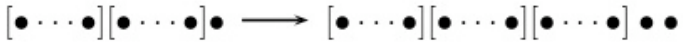
Des noirs

Maintenant les cailloux sont tous de la même couleur, mais on regarde les alignements plus globalement, [4]. Quand on avait des couleurs, on pouvait regarder des configurations comme *trois rouges — trois bleus*, maintenant on examine tout l'alignement et on regarde si on peut faire des regroupements, par exemple, deux paquets de même taille (premier cas) ou deux paquets presque de même taille (deuxième cas). Dans le deuxième cas, je veux dire par « presque » qu'on a deux paquets de même taille plus un caillou restant [5]. Supposons qu'il y ait deux règles :

La terminaison est facile et je suppose que vous voyez pourquoi le processus ne peut pas « diverger ».

Considérons maintenant deux règles plus difficiles :





qui veulent dire que si on peut rassembler les cailloux en deux groupes [6] de même taille, la transformation enlève un de ces groupes, et si au contraire on ne peut pas assembler les cailloux en deux groupes de même taille, alors on peut les assembler en deux groupes de même taille avec un caillou supplémentaire, dans ce cas la transformation ajoute un troisième groupe de la taille des deux premiers et adjoint un deuxième caillou supplémentaire. On va donc de temps en temps ajouter des cailloux, de temps en temps en retirer. Va-t-on s'arrêter ? La réponse est aujourd'hui inconnue [7].

Les mathématiques comme un jeu de cailloux [8] ?

Les mathématiques utilisent une méthode unique parmi les autres sciences. Tous les énoncés (les propositions, les théorèmes etc.) doivent être rigoureusement justifiés. Pour cela, il faut construire une « démonstration » qui consiste en une suite d'énoncés intermédiaires dont chacun est obtenu à partir des précédents en utilisant un certain nombre de « règles logiques » bien établies, du genre « *si j'ai ceci alors j'ai ça* » que j'enchaîne avec, par exemple, « *si j'ai A et si j'ai B alors j'ai "A et B"* ». Bien sûr, il faut bien partir de quelque part : ce sont des énoncés bien précis qu'on appelle les *axiomes*. On peut donc penser l'édifice mathématique comme un immense jeu, un peu comme un jeu d'échecs : il y a une position initiale (analogue aux axiomes), il y a des règles de mouvement des pièces (analogues aux règles de logique) et il s'agit de mouvoir les pièces (en respectant les règles !) pour aboutir à certaines positions (les théorèmes). Voici un jeu « mathématique » :

- Votre mission est de démontrer que tout nombre pair (autre que 2) est la somme de deux nombres premiers.
- Vous ne pourrez compter que sur les axiomes et les règles.
- À vous de naviguer pour arriver jusqu'au but.
- Bonne chance !

Ce jeu solitaire est beaucoup plus difficile que le jeu d'échecs !

Un peu plus de détails sur les démonstrations : cliquer pour déplier.

> *Le système logique appelé déduction naturelle est fondé sur la notion de séquent. Un séquent s'écrit $\Gamma \vdash p$ où p est une proposition et Γ est une ensemble de propositions, le séquent $\Gamma \vdash p$ signifie « à partir des hypothèses de l'ensemble Γ, je peux démontrer la proposition p ». Un séquent dans lequel $p \in \Gamma$ nous sert à démarrer nos démonstrations, en effet, cela signifie que « p figure parmi les hypothèses », on dit aussi que c'est un axiome. Nous allons présenter les démonstrations verticalement. Chaque ligne représente un séquent avec sa justification : ou bien c'est un axiome, ou bien c'est la conséquence de l'emploi d'une règle de déduction (voir ci-dessous) qui utilise des séquents déjà démontrés, dans des lignes précédentes. Un séquent où Γ est vide, ce que l'on écrit traditionnellement $\vdash p$, et qui est la dernière ligne d'une démonstration est appelé un théorème. Voici deux règles qui traitent du symbole logique d'implication, noté $\Rightarrow$:*
>
> - ***règle 1*** *$\Gamma \vdash q$ est justifiable par $\Gamma \vdash p \Rightarrow q$ et $\Gamma \vdash p$.*
> - ***règle 2*** *$\Gamma \vdash p \Rightarrow q$ est justifiable par $\Gamma, p \vdash q$.*
>
> *Voici une démonstration du théorème $\vdash (p \Rightarrow (q \Rightarrow r)) \Rightarrow (q \Rightarrow (p \Rightarrow r))$, (qui énonce un certaine forme de commutativité : dans une implication je peux permuter les propositions qui sont devant le signe $\Rightarrow$) :*
>
> - *1. $p \Rightarrow (q \Rightarrow r), q, p \vdash p \Rightarrow (q \Rightarrow r)$ axiome*
> - *2. $p \Rightarrow (q \Rightarrow r), q, p \vdash p$ axiome*
> - *3. $p \Rightarrow (q \Rightarrow r), q, p \vdash q \Rightarrow r$ par la règle 1 appliquée aux lignes 1 et 2*
> - *4. $p \Rightarrow (q \Rightarrow r), q, p \vdash q$ axiome*
> - *5. $p \Rightarrow (q \Rightarrow r), q, p \vdash r$ par la règle 1 appliquée aux lignes 3 et 4*
> - *6. $p \Rightarrow (q \Rightarrow r), q \vdash p \Rightarrow r$ par la règle 2 appliquée à la ligne 5*
> - *7. $p \Rightarrow (q \Rightarrow r) \vdash q \Rightarrow (p \Rightarrow r)$ par la règle 2 appliquée à la ligne 6*
> - *8. $\vdash (p \Rightarrow (q \Rightarrow r)) \Rightarrow (q \Rightarrow (p \Rightarrow r))$ par la règle 2 appliquée à la ligne 7.*

Ceux qui ont déjà fait une démonstration mathématique ont déjà éprouvé cette insatisfaction devant une

démonstration qui ne leur plaisait pas et qu'ils voulaient transformer, soit pour la rendre plus belle ou plus directe, soit pour éviter un argument qui leur semblait n'avoir rien à faire avec le théorème à démontrer. Vous avez trouvé avec beaucoup de peine l'échoppe que vous cherchiez dans le souk de Marrakech et vous voulez expliquer le chemin à un ami : il faut qu'il soit le plus simple possible et sans détour inutile. Est-ce possible ?

Une démonstration, avec sa suite d'énoncés intermédiaires, utiles ou inutiles, est similaire à nos rangées de petits cailloux, du premier au dernier, de l'axiome au théorème... Transformer une démonstration, pour en trouver une meilleure ressemble à transformer un alignement de petits cailloux. On voit donc que nos petits jeux de cailloux [**9**] ont une importance fondamentale en mathématiques, puisqu'ils décrivent la structure même de l'activité du mathématicien.

En fait, les logiciens ne considèrent pas seulement les démonstrations comme quelque chose qui sert aux mathématiciens, mais aussi comme un véritable objet mathématique au même titre que les nombres ou les figures géométriques. Les objets que manipulent les logiciens sont les démonstrations : ils sont donc amenés à faire des démonstrations à propos des démonstrations... Pourquoi pas ?

L'édifice des mathématiques est-il solide ?

Nous allons voir comment la terminaison peut garantir sa solidité.

On dit qu'une théorie est cohérente si on ne peut pas y démontrer tout et son contraire [**10**] ou ce qui revient au même si on ne peut pas y démontrer la proposition contradictoire (la proposition « faux »). Il est évidemment essentiel de démontrer que la logique est cohérente, sinon c'est tout l'édifice des mathématiques qui s'effondre.

On voit donc combien les petits cailloux sont importants !

Un peu plus détail sur la cohérence : cliquer pour déplier.

> ***L'élimination des coupures***
>
> *La démonstration de la cohérence proposée par* ***Gerhard Gentzen*** *repose sur le principe d'élimination des coupures. Ce que les logiciens à la suite de Gentzen appellent une coupure est la méthode qui consiste à utiliser un théorème intermédiaire pour démontrer un théorème principal. Si pour démontrer le théorème principal* ***T*** *je fais appel au théorème intermédiaire* ***t*** *je fais une coupure, ce qui signifie que d'une part je démontre le théorème t et que d'autre part je démontre* ***T*** *en utilisant dans mes hypothèses le théorème* ***t*** *. Éliminer cette coupure c'est proposer une démonstration qui se passe du théorème t. Pour cela, Gentzen propose une méthode qui consiste à transformer la démonstration en supprimant une coupure après l'autre, quitte à utiliser des coupures à des niveaux supposés [11] plus bas dans la démonstration. Ce principe d'élimination des coupures fonctionne comme les méthodes que nous venons de voir, mais ici on « transforme » des démonstrations en éliminant des coupures et la clé de la réussite est de montrer que le procédé termine. Si c'est le cas, on peut transformer par élimination de toutes ses coupures toute démonstration en une démonstration sans coupure (sans théorème intermédiaire). En regardant les règles de démonstration [12] on voit qu'une démonstration sans théorème intérmédiaire satisfait alors la propriété de la sous-formule, c'est-à-dire que c'est une démonstration qui n'utilise que des sous-formules*

de la proposition que l'on cherche à démontrer.

La démonstration de la cohérence

Supposons que le système de déduction soit incohérent, donc supposons qu'il y existe une démonstration de faux. Par élimination des coupures, il existe une démonstration de faux qui satisfait la propriété de la sous-formule, c'est-à-dire une démonstration de faux qui n'utilise que des sous-formules de faux, mais comme faux n'a pas de sous-formule, une telle démonstration de faux sans coupure n'existe pas, donc il n'y a aucune démonstration de faux. Donc si j'ai pu démontrer l'élimination des coupures (comme une démonstration de terminaison), le système est cohérent.

Les calculs et l'indécidabilité de la terminaison

Pour définir une fonction ou une technique de simplification d'expressions, on utilise un procédé qui s'apparente à celui des petits cailloux [13], mais pour garantir que le programme ou la simplification est correct, il faut s'assurer que le procédé (ou le programme d'ordinateur), qui définit la fonction ou la simplification, s'arrête [14]. *C'est tout simple !* Il suffit d'écrire un programme général qui prend comme donnée le programme dont on veut garantir la terminaison et qui teste si le dit programme termine ou non. Eh bien non, ça n'est pas si simple que ça, parce qu'*Alan Turing* a montré qu'un tel programme universel, qui teste la terminaison, n'existe pas. Cela a comme conséquence que pour chaque programme (ou chaque famille de programmes) il faudra inventer une méthode de terminaison ad hoc. Mais il restera des programmes pour lesquels toutes les méthodes ad hoc connues échouent lamentablement [15].

Des logiciels pour vérifier la terminaison

Comme tester la terminaison est assez subtil, nécessite des batteries de méthodes assez différentes les unes de autres et peut servir dans des programmes informatiques où la vie de personnes est en jeu [16], les scientifiques ont écrit des logiciels pour faire ce travail.

Une compétition

Comme les méthodes sont ad hoc, tous les logiciels n'ont pas le même comportement et on aimerait savoir quel est le meilleur logiciel. Par exemple on aimerait savoir si le logiciel de l'Université d'Orsay bat celui de l'Université d'Aix la Chapelle ou celui de l'Université d'Innsbrück ou celui de celle de Leipzig, d'Eindhoven ou de Nancy. Sans vouloir les positionner dans un ordre strict [17] on peut vouloir savoir sur quel type d'exemple tel ou tel logiciel est meilleur que tel autre. Pour départager tout le monde, des chercheurs ont eu l'idée de mettre sur pied une compétition où les logiciels de terminaison s'affronteraient sur des échantillons d'exemples bien choisis et renouvelés [18]. De cette compétition doivent sortir défis et améliorations.

Des démonstrations certifiées

Croire qu'un logiciel peut effectivement garantir la terminaison reste un acte de foi que tous ne veulent pas faire [19], aussi demande-t-on aujourd'hui aux logiciels de fournir un certificat, c'est-à-dire le terme d'une démonstration formelle qui peut être vérifiée par un outil adéquat indépendant [20] ou par un mathématicien courageux ! Aujourd'hui on peut espérer que plus personne ne mette en doute une démonstration de terminaison automatisée qui fournit un certificat.

Une terminaison très lente : cliquer ici pour déplier

Voici un processus qui termine très très lentement.

Nous allons considérer des suites qui sont construites de la façon suivante : si le nombre en question n'est pas 0, on commence par retrancher 1 puis on fait croître la base [21]. Soyons plus précis et considérons tout d'abord un exemple. Prenons le nombre 111 (cent onze en numération décimale) et écrivons-le en base 3, remarquons, qu'il s'agit du nombre quatre vingt un, plus vingt sept, plus trois, soit 81+27+3, autrement dit si l'on écrit aussi les exposants [22] en base 3, cela donne 3^4 + 3^3 + 3. Soyons extrémistes et écrivons tout en base 3, c'est-à-dire les exposants compris, cela donne : 3^(3^(3^0) + 3^0) + 3^(3^(3^0)) + 3^(3^0), mais nous écrirons plutôt 3^(3+1) + 3^3 + 3 [23]. Passer en base 4 consiste à remplacer tous les 3 par des 4, ce qui donne 4^(4+1) + 4^4 + 4, autrement dit 1284 en numération décimale.

Examinons maintenant le processus de construction de ces suites qui terminent très lentement, on commence par une décomposition en base 2, puis une décomposition en base 3, 4, 5 etc. On rappelle qu'on passe d'un nombre n(k) de la suite au suivant en lui retirant 1, ce qui donne m(k), et en changeant de base, ce qui donne n(k+1). Commençons par n(2)= 15 on a alors

$$
\begin{array}{lcll}
n_2 & = & 2^{2+1} + 2^2 + 2 + 1 & = 15 \\
n_3 & = & 3^{3+1} + 3^3 + 3 & = 111, \\
n_4 & = & 4^{4+1} + 4^4 + 2 & = 1\,282, \\
n_5 & = & 5^{5+1} + 5^5 + 1 & = 18\,751, \\
n_6 & = & 6^{6+1} + 6^6 & = 326\,592, \\
n_7 & = & 7^{7+1} + \ldots & = 5\,764\,801 + \ldots, \\
 & \vdots & & \\
n_{40000} & = & 40000^{40000+1} + \cdots &
\end{array}
$$

Cette suite [24] semble diverger et on se dit que retirer 1 compte pour bien peu de chose comparé au changement de base. Eh bien il n'en est rien et quel que soit le nombre dont on part, on arrive tôt ou tard à 0, auquel cas on s'arrête. Mais cette démonstration de terminaison a cette particularité, démontrée par les logiciens L. Kirby et J. Paris [25], qu'elle ne peut pas être faite dans l'arithmétique élémentaire. Ceci conforte un résultat bien connu, appelé théorème d'incomplétude de Gödel, qui dit que tout ne peut pas être fait dans l'arithmétique élémentaire.

P.S. :

Encore très active, la recherche sur la terminaison n'est pas près de s'arrêter.

Notes

[▲1] La réponse est oui, mais sa solution sortirait du cadre de cet exposé, une étude complète de ce type de réduction a été faite par **A. Geser et H. Zantema**.

[▲2] Le sens de *« terminaison »* utilisé ici est un emprunt de l'anglais, il signifie le fait qu'un processus s'arrête ou non.

[▲3] **Là encore la réponse est oui, mais n'est pas évidente**.

[▲4] Il ne s'agit plus d'une *petite main*, mais d'une *grosse main* !

[▲5] En fait on teste la parité ou l'imparité, mais on a besoin de savoir la taille des paquets pour la reporter après transformation.

[▲6] non vides

[▲7] Ce **problème** est une présentation de la conjecture connue sous le nom de *problème 3x+1*, de *problème de Collatz* ou de *problème de Syracuse* qui est toujours non résolue à ce jour. Il s'appelle *problème 3x+1* parce qu'il est souvent présenté sous la forme suivante : *« prenez un nombre x, s'il est pair divisez le par 2, s'il est impair et n'est pas égal à 1 alors multipliez le par 3 et ajoutez 1 (autrement dit remplacez le par 3x+1), puis recommencez »*.

[▲8] Notez que les mots calculs et cailloux ont la même origine

[▲9] et leurs petites sœurs les démonstrations !

[▲10] Autrement dit, on ne peut pas y démontrer *A* et non *A*.

[▲11] je dis « supposées », parce que dire qu'un théorème intermédiaire est « plus bas » qu'un autre est lié à la terminaison !

[▲12] Cela pourra faire l'objet d'un autre article.

[▲13] On dit qu'on définit une machine abstraite.

[▲14] La terminaison des programmes ou de ce qui leur ressemble s'appelle aussi la *normalisation forte*.

[▲15] C'est le cas de la fonction *3x+1*.

[▲16] Imaginez un programme qui fait un certain nombre de tests avant d'activer le freinage d'une automobile.

[▲17] Les chercheurs savent bien qu'un ordre total n'est pas possible, comme ce serait folie de dresser un **ordre total des universités** !

[▲18] Voyez pour cela le **site** de la compétition de terminaison.

[▲19] A commencer par l'auteur de cet article, qui ne croit pas plus le **Sar Rabindranath Duval** incarné par Pierre Dac.

[▲20] voyez le site du **groupe de travail sur la terminaison certifiée**

[▲21] La notion de base est rappelée dans l'article ***Et si les nombres pouvaient être infinis à gauche de la virgule plutôt qu'à droite…***

[▲22] Je note les puissances par l'opérateur binaire ^, autrement dit *3^3* veut dire *3 fois 3 fois 3*.

[▲23] car nous savons que *3^0* c'est *1* et que *3^1* ou *3^(3^0)* c'est *3*.

[▲24] que l'on appelle *suite de Goodstein*

[▲25] L. Kirby et J. Paris, « Accessible independence results for Peano arithmetic », dans *Bull. London. Math. Soc.*, 14 (1982), 285-93

Crédits images

Pour citer cet article : **Pierre Lescanne**, **Est-ce que ça s'arrête ?**. *Images des Mathématiques*, CNRS, 2009. En ligne, URL : `http://images.math.cnrs.fr/Est-ce-que-ca-s-arrete.html`